BOLT BERANEK AND NEWMAN INC

CONSULTING • DEVELOPMENT • RESEARCH

Report No. 2175 July 1971

INTERFACE MESSAGE PROCESSORS FOR THE ARPA COMPUTER NETWORK

QUARTERLY TECHNICAL REPORT NO. 10
1 April 1971 to 30 June 1971

Principal Investigator: Mr. Frank E. Heart
Telephone (617) 491-1850, Ext. 470

Sponsored by
Advanced Research Projects Agency
ARPA Order No. 1260

Contract No. DAHC15-69-C-0179
Effective Date: 2 January 1969
Expiration Date: 31 December 1971
Contract Amount: $3,927,787

Title of Work: IMP

Submitted to:

Director
Advanced Research Projects Agency
Washington, D.C. 22209

Attention: Dr. L.G. Roberts
Telephone (202) OX4-5922

INTERFACE MESSAGE PROCESSORS FOR THE ARPA COMPUTER NETWORK

QUARTERLY TECHNICAL REPORT NO. 10
1 April 1971 to 30 June 1971

Submitted to:

Advanced Research Projects Agency
Washington, D.C. 22209
Attn: Dr. L.G. Roberts

This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract No. DAHC-15-69-C-0179.

TABLE OF CONTENTS

1. INTRODUCTION

This Quarterly Technical Report, No. 10, describes aspects of ~~our~~ work on the ARPA Computer Network during the second quarter of 1971.

Our efforts during this period were devoted primarily, though not exclusively, to the Terminal IMP. The quarter saw the ARPA Network remain stable in size; we used this opportunity to study traffic in the net, to plan and implement improvements in the operation of the Network Control Center at BBN, to discover and correct some minor bugs in the operational IMP hardware and software, and to continue investigation of routing problems.

During the quarter our participation in the Network Working Group increased substantially, with several members of the staff devoting significant amounts of time to Working Group subcommittees and to the preparation of Network documents. Some of our activities in this area are described in Section 2.

Work on the Terminal IMP progressed on several fronts. By the end of the quarter we had received three additional 316s from Honeywell. Two Multi-Line Controllers are in final assembly and three more are under construction. A variety of terminal devices have now been tested with the Terminal IMP hardware and prototype software. Significant portions of the Terminal IMP software have been written and partially debugged, and the Terminal IMP command language has been largely specified. By the end of the quarter a terminal connected to the prototype Terminal IMP had successfully logged into the BBN TENEX system. All of these developments are covered in some detail in Section 3.

2. NETWORK WORKING GROUP PARTICIPATION

Early in 1969 the Network Host organizations formed a Network Working Group (NWG), under the chairmanship of Steve Crocker, to facilitate the exchange of ideas about network use and to formulate specifications for Host-to-Host communications. Although BBN has participated in the activities of the NWG since its inception, during the last two quarters we have significantly expanded our participation, and have been particularly involved in the following areas:

1) Host-to-Host Protocol: The so-called Host-to-Host Protocol was the first protocol developed by the NWG. It is primarily concerned with the method of establishing process-to-process "connections" over the network, and with the control of data flow over these connections. (It is not generally concerned with the syntax or semantics of the data being transferred over the connections.) The first "official" specification of this protocol was published in mid-1970; by early 1971 various problem areas of the original specification were becoming apparent. We have participated in a committee of the NWG which was formed to revise the protocol; we also assumed responsibility for formalizing and publishing a revised protocol specification.

2) Resource Notebook: If the network is actually to be used for resource sharing it is essential that each Host organization have convenient access to a description of the resources available at other sites. With this in mind we have undertaken the preparation of a document entitled the "ARPA Network Resource Notebook" which is

intended to provide summary information for each Host site, as well as references to more detailed site documentation. We prepared an outline of the summary information desirable, and requested each site to supply that information directly to us. The information was edited into a uniform format at BBN, and the original Resource Notebook (representing 13 Hosts) was released via the Network Information Center at SRI in April of this year. We have just completed work on the first revision of the Notebook which now provides summary information for a total of 17 (of 19) Host organizations.

3) TELNET Protocol: The TELNET protocol specifies a method for making a terminal (or process) at a "using" site appear, to the system or to a process at a "serving" site, logically equivalent to the type of terminal normally used at the serving site. This is accomplished by specifying the characteristics of a "Network Virtual Terminal" (NVT); each Host is responsible for mapping between locally-used conventions (character codes, echo modes, etc.) and NVT conventions. We have been heavily involved in the committee responsible for the specification of TELNET, since Terminal IMPs will most commonly operate under this protocol.

4) Data and File Transfer Protocols: We have also been represented on the committee of the NWG responsible for proposing protocols for data transfer and file transfer. As with TELNET, the protocols finally adopted by the NWG will significantly affect the performance of the Terminal IMPs.

In addition to the specific areas described above, we have somewhat expanded our participation in the quarterly NWG meetings, and have undertaken some specific small tasks, such as determining and summarizing site status, at the request of the Network Working Group.

3. TERMINAL IMP

3.1 Hardware Development

Early in the last quarter we reviewed aspects of the Terminal IMP software requirements including device buffering for medium and high speed terminals, character code translation, and the various Host-to-Host protocols with which the Terminal IMP must comply. This review made clear to us that the originally planned 16K memory capacity would be insufficient, and the decision was made to upgrade the memory capacity to 20K. The additional 4K memory module will physically occupy space at the top of the highboy rack formerly allocated to modem and Host interfaces. Thus, the initial Terminal IMP will be physically limited to interfacing either with three modems or with two modems and one Host. We intend to provide a version of the Terminal IMP with an additional half-rack which will modify these limitations.

Memory retrofits have been ordered for the prototype Terminal IMP and the Terminal IMP scheduled for field installation, both of which were delivered to BBN some time ago. During this quarter we took delivery of three additional H-316 Terminal IMP computers from Honeywell; these machines are all equipped with the full 20K memories. Construction then proceeded through the rest of the quarter on all of the first four deliverable Terminal IMPs, with the first delivery scheduled for the middle of the third quarter of 1971. Work centered on detailed documentation of mechanical and electrical assemblies, selection and testing of components and assemblies, selection of qualified vendors, and final mechanical and electrical design cleanup.

The prototype Multi-Line Controller (MLC) common logic unit and 40 Line Interface Units (LIUs) were completed and checked out. In particular, the MLC has run successfully at all speeds up to and including 19.2 kilobits and with several combinations

of mixed speeds. The MLC has been tested with LIUs in each of the 64 possible slots, although for complete testing we must await the fabrication of additional LIUs. Using a prototype test program we have operated the MLC at data rates well in excess of the Terminal IMP design goal (100 Kilobits total) and observed gradual, rather than catastrophic, degradation.

A second MLC has been fabricated and partially tested in the same manner as the prototype, and three additional MLCs were under construction by the end of the quarter. In addition, a test program for the MLC was partially completed and run. This test program will be used in the production of MLCs as an analog to IMPTEST in the production of IMPs.

3.2 Terminal Checkout

During the last quarter, the following terminals were received and tested with the BBN Multi-Line Controller and a prototype test program.

ODEC Line Printer and Line Adapter
Execuport Terminal
IMLAC PDS-1 Graphics Terminal
TTY Model 33
Infoton Alphanumeric CRT Terminal Vista 1-H
IBM 2741 Communications Terminals

ODEC

The ODEC line printer is a 200-line-per-minute, 132-column tabletop-size chain printer. It prints a 64-character subset of ASCII. In order to provide the capability of operating as a stand-alone remote terminal at the end of a dial-up telephone line, a

Communications Line Adapter was ordered from ODEC with the printer. The adapter provides a 256-character buffer to allow for instantaneous differences between the character input and character printing rates which occur because of line length variations. The 200-line-per-minute printing rate, if full 132-character lines are used, is equivalent to an asynchronous data rate of approximately 4600 bits per second. In practice, since full 132-character lines are not always used, the rate of successive lines must not exceed approximately 3 per second (200 lines/minute) long term average or the Line Adapter buffer will overflow and data will be lost. In the event that the buffer nears overflow (16 to 31 character spaces left) a reverse channel signal is furnished by the adapter. This signal will be used to temporarily halt the flow of data from the MLC computer program.

The printer has been operated at 1800 bits/second, but the rate of the line adapter may be strapped to 600, 1200, or 2400 bits/second, if desired. A rate of 1200 is anticipated for use over the dial-up phone system using 202C type modems and this mode will be tested shortly.

Execuport

The Execuport 300 is a stand-alone portable communications terminal with a built-in acoustic coupler for use over the dial-up telephone system. It uses ASCII characters at rates of 110, 150, and 300 bits/second, determined by a switch set by the operator. Printout is a 5 x 7 dot matrix on thermal-sensitive paper (single copies only).

IMLAC PDS-1

The IMLAC PDS-1 is an intelligent (i.e., it contains a computer) terminal which is designed for graphics uses. The computer has a 16-bit word and our version has 4K of core. The terminal may be used for alphanumerics if desired. The set of characters used depends on the particular program used in the computer. In the tests of PDS-1 with the MLC, the standard IMLAC edit program was used (implies ASCII characters).

The transfer data rate of the PDS-1 is defined by strapping and components on a printed circuit card. We have operated at 110 bits/second, 1800 bits/second, and 9600 bits/second.

Infoton Vista 1-H

This is an alphanumeric-only CRT terminal which uses asynchronous ASCII characters. The data rate is switch selectable from 110 bits/second to 4800 bits/second. The device has 80 columns and 20 lines. It has been tested at all its switched rates with the MLC.

IBM 2741

Two versions of IBM 2741 were leased from IBM for testing with the MLC, a Correspondence code version and a PTTC/EBCD code version. These terminals have a rate of 134.5 bits/second and are half duplex only. A reverse break option will soon be installed in the terminals and tested with the MLC. To date, these terminals present the largest problems in operation because of the code sets used (both non-ASCII) and also the complicated handshaking procedure necessary to control the terminals.

No special problems were encountered with any of the terminals in testing. All MLC rates from 110 baud to 19.2 kilobaud were exercised by various combinations of terminals. Several of the terminals were also exercised by connecting them to the BBN TENEX System. Telephone company modems of the 103A and 202C type were installed in the IMP room and tests were conducted to insure their compatibility with the MLC. Once again no significant problems arose and several terminals were connected to the MLC and operated using these modems.

3.3 Terminal IMP Control Language

The prototype Terminal IMP was successfully placed on the ARPA Network and a user was able to log into BBN TENEX using a prototype of the Terminal IMP program. The function of this program is to transmit data, primarily characters, from a terminal to some remote Host and to return the Host's response; this activity is expected to be part of an interactive dialogue in most cases. To perform this function the Terminal IMP software needs to know the values of several parameters for each device; the user will be expected to provide these parameters via the Terminal IMP control language, which is described below. It should be noted that this control language is still under development and subject to change, but is expected to be essentially as described here at the time of delivery of the first Terminal IMP.

As the Terminal IMP passes characters from a terminal to the net it briefly investigates them to see if they are one of three special characters which require more examination. These characters are @, LINEFEED, and END-OF-MESSAGE. (A "delete last character" character and a "cancel line" character are expected to be added in the future.) It should be emphasized here that the

interpretation of these special characters depends on mode parameters, and that it is possible to set the modes so that the characters are not interpreted at all. In this latter mode, the terminal passes straight binary to the Host.

LINEFEED and END-OF-MESSAGE may be used to signal the IMP that it is time to pack the characters it has been accumulating into a message and dispatch them to the Host. This option might be used in line-at-a-time operation. The special characters will be included at the end of the message. See the TRANSMIT command (below) for additional details.

The @ signals the beginning of a command to the Terminal IMP. It may occur anywhere in the input. All the characters from the @ to the command terminator will be interpreted as a command to the IMP. They will not be put in the message buffer and the remote Host will never see them. The command terminator is normally the pair "CARRIAGE-RETURN - LINEFEED". The only exception to this rule is the command @@ which puts a single @ in the regular output buffer. This exception allows one to send this special character on to the Host if @ is part of its control set. (This has nothing to do with sending binary, however.)

The regular commands consist of the following elements in the indicated order:

@	required
<DEVICE NUMBER>	optional
<COMMAND TYPE>	required
<NUMERIC PARAMETER>	sometimes required
CARRIAGE-RETURN	required
LINEFEED	required

An example of a regular command is

@ RECEIVE FROM HOST 6 CARRIAGE-RETURN LINEFEED

The <DEVICE NUMBER> specifies which device the user is setting parameters for. Usually, it will be his own device, in which case it may be omitted. However, it will be necessary to set up parameters for devices like card readers, tape drives, etc. which cannot set their own modes. Such an option requires protection, which will take the following form:

1) The executive teletype can change anyone's parameters.

2) Certain devices (such as line printers) can be captured by another device if they are in the free state. Once captured only the capturing device can set and change their parameters until they are freed.

In what follows the device number will be omitted from all examples and discussion unless it plays an unusual role. The command terminator will be omitted as well.

The <COMMAND TYPE> consists of one to four English words. These words are carefully chosen so that commands can be uniquely distinguished by the first letter of each word. This has two benefits: the IMP can save valuable core by only investigating the first letter after each space (or set of spaces), and a fluent user can abbreviate his commands; the example above becomes

@ R F H 6

Commands to Set Parameters for Connections

@ HOST #
@ SEND TO SOCKET #
@ SEND ON LINK #
@ RECEIVE FROM HOST #
@ RECEIVE FROM SOCKET #

These commands set parameters for the terminal in preparation for establishing communications with a remote process. The establishment of the parameters does not initiate the protocol to establish the connection, but it will result in the exchange of protocol messages to close a previously existing connection. A fundamental system constraint is that each device participates in only a single connection at a time.

The HOST command is shortened because it is part of the standard TELNET login sequence.

The parameter-setting commands do not permit establishing the receive link since, by convention, the receive link will always be the device number plus two. Similarly, the local socket numbers will be a fixed multiple of the device number, with the "gender" bit set appropriately.

The Terminal IMP initially ignores the niceties of protocol and simply prints any incoming message on the device indicated by the link. Establishing RECEIVE parameters amounts to locking out messages from all but the authorized source. The Terminal IMP also initially ignores protocol on the transmit side, permitting the user to establish the transmit link. (Protocol requires the RECEIVE side of a connection to specify the link.)

The Terminal IMP imposes only the constraint that a device will not be allowed to establish the same Host and link already used by another device. The commands described here allow complete manual specification of a connection; however, the various protocol options to be described later will automatically supply several of the parameters.

It is reasonable to also have an "unspecified" state for the transmit parameters to protect against operator error. For example, when a user changes the remote Host #, the link value will automatically be set to "unspecified".

Commands to Control Transmission

@ TRANSMIT EVERY CHARACTER
@ TRANSMIT EVERY #
@ TRANSMIT ON END-OF-MESSAGE
@ TRANSMIT ON LINEFEED
@ TRANSMIT ON NO CHARACTER
@ TRANSMIT NOW

The Terminal IMP needs to know how many characters to accumulate before trying to send off a message. The user has three basic options: he can specify a count of characters (EVERY CHARACTER amounts to a count of one). If he doesn't specify a count, the IMP supplies some large number. The user can also specify termination on either of two special characters (END-OF-MESSAGE and LINEFEED). He may set both or neither (NO CHARACTER) if he wishes. The third option is to specify the end of each message manually (TRANSMIT NOW). However, the IMP may be unable to send the message at the time specified by the user. When this happens, the message will be sent at the next opportunity, and will include *all* the characters received up to that time. This is in line with the Host protocol philosophy which asserts that message boundaries should have no particular significance.

Commands to Set Echo Mode

@ ECHO ALL
@ ECHO NONE
@ ECHO HALF-DUPLEX

These commands tell the Terminal IMP what characters to echo to the device. ALL causes all characters input from the terminal to be echoed (full duplex with local echoing). NONE causes none of the characters which are passed to the Host to be echoed (full duplex with remote echoing); however, all characters which are intercepted by the Terminal IMP are echoed. HALF-DUPLEX causes the Terminal IMP to

- echo nothing
- ignore input while output (to the terminal) is in progress
- prevent output while input (from the terminal) is in progress

Commands to Transmit Special TELNET Codes

@ SEND SYNC
@ SEND BREAK

The TELNET protocol includes the definition of two special characters which must be under the control of the user. The "sync" character is inserted in the data stream when an "Interrupt" signal is sent on the "control link" from one Host (in this case the Terminal IMP) to another; it allows the receiving Host to synchronize the data stream with the control stream. The "break" character is used to gain the attention of the serving Host.

Commands to Activate Terminal IMP Control

@ INTERCEPT ESCAPE

@ INTERCEPT NONE

These commands determine whether the special "Escape Character" @ (and, in the future, other special characters) will be intercepted by the Terminal IMP. In ESCAPE mode it is intercepted; in NONE mode it is not. The user must be careful with the NONE command, as it can disconnect him from his Terminal IMP. It is necessary, however, for devices like card readers.

Commands to Set Device Parameters

@ DEVICE RATE #

@ DEVICE CODE ASCII

@ DEVICE CODE EBCDIC

@ DEVICE SIZE #

These commands set up device rate, code conversion, and character size. They will normally be executed remotely since they must be known before the local device could execute the command. The numerical parameter must be obtained by the user from a set of (hard copy) tables which will be supplied with the Terminal IMP. The ASCII and EBCDIC commands indicate standard 7-bit codes in an 8-bit field.

Command to Release a Device

@ # GIVE BACK

A device such as a line printer can be captured by a terminal through the process of setting device parameters for it. Only one terminal can be in control of a device at a time; therefore, the terminal user should release it when finished. As previously mentioned, most terminals are under their own command and cannot be captured. A command attempting to set up parameters on a device already captured will be answered with an error message.

Commands to Set Linefeed Mode

@ FEED AUTO
@ FEED MANUAL

When in AUTO mode, the Terminal IMP will generate a LINEFEED each time it receives a CARRIAGE-RETURN from the terminal. The LINEFEED will be transmitted both to the terminal (regardless of the ECHO mode) and to the receiving Host. One of the consequences of AUTO mode is that the terminal user terminates commands by typing *only* CARRIAGE-RETURN.

Commands to Follow Standard Protocols

@ PROTOCOL TO TRANSMIT
@ PROTOCOL TO CLOSE TRANSMIT
@ PROTOCOL TO RECEIVE
@ PROTOCOL TO CLOSE RECEIVE
@ PROTOCOL BOTH
@ LOGIN
@ CLOSE

These commands instruct the Terminal IMP to follow standard Host-to-Host protocol procedures. The pair of TRANSMIT commands attempt to open or close a connection for which the Terminal IMP is the sender. Similarly, the pair of RECEIVE commands attempt to open or close a connection for which the Terminal IMP is the receiver. The BOTH command attempts to open both a send and a receive connection. Note that the Host number and the remote socket number(s) must be set before using these commands.

The LOGIN command instructs the Terminal IMP to follow the standard Initial Connection Protocol. The CLOSE command attempts to close <u>both</u> a send and a receive connection. It may be used either after a successful LOGIN or after establishing both connections manually.

During the third quarter of 1971 we will install the first three operational Terminal IMPs. We anticipate that as users gain experience with the command language, some tailoring will prove desirable. We also expect to acquire and test other types of terminal devices with the MLC. In addition, we will continue to develop and improve the operation of the Terminal IMP software.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
Bolt Beranek and Newman Inc. 50 Moulton Street Cambridge, Mass. 02138	UNCLASSIFIED
	2b. GROUP

3. REPORT TITLE

QUARTERLY TECHNICAL REPORT NO. 10

1 April 1971 to 30 June 1971

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)

Bolt Beranek and Newman Inc.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
July 1971	18	

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
DAHC15-69-C-0179	BBN Report No. 2175
b. PROJECT NO. 1260	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Advanced Research Projects Agency Washington, D.C. 22209

13. ABSTRACT

The basic function of the IMP computer network is to allow large existing time-shared (Host) computers with different system configurations to communicate with each other. Each IMP (Interface Message Processor) computer accepts messages for its Host from other Host computers and transmits messages from its Host to other Hosts. ~~Since~~ there will not always be a direct link between two Hosts that wish to communicate, individual IMPs will, from time to time, perform the function of transferring a message between Hosts that are not directly connected. This then leads to the two basic IMP configurations — interfacing between Host computers and acting as a message switcher in the IMP network. The message switching is performed as a store and forward operation. Each IMP adapts its message routine to the condition of those portions of the IMP network to which it is connected. IMPs regularly measure network performance and report in special messages to the network measurement center. Provision of a tracing capability permits the net operation to be studied comprehensively. An automatic trouble reporting capability detects a variety of network difficulties and reports them to an interested Host. An IMP can throw away packets that it has received but not yet acknowledged, transmitting packets to other IMPs at its own discretion. Self-contained network operation is designed to protect and deliver messages from the source Host to the destination IMP.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

UNCLASSIFIED
Security Classification

A-31408

UNCLASSIFIED

Security Classification

14 KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Computers and Communication						
Store and forward communication						
ARPA Computer Network						
Honeywell DDP-516						
Honeywell DDP-316						
IMP						
Interface Message Processor						
Multi-Line Controller (MLC)						
Terminal IMP						

DD FORM 1 NOV 65 1473 (BACK)

UNCLASSIFIED

Security Classification